UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/964,591	09/28/2001	Akio Matsuda	1075.1174	2406

21171 7590 09/01/2005

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
THANGAVELU, KANDASAMY	
ART UNIT	**PAPER NUMBER**
2123	

DATE MAILED: 09/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/964,591	MATSUDA ET AL.	
	Examiner	Art Unit	
	Kandasamy Thangavelu	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 May 2005 and 29 July 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 September 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *28 September 2001*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This communication is in response to the Applicants' Response mailed on May 10, 2005 and July 29, 2005. Claims 1, 4-7 and 12-41 were amended. Claims 1-41 of the application are pending. This office action is made final.

Claim Objections

2. The following is a quotation of 37 C.F.R § 1.75 (d)(1):

> The claim or claims must conform to the invention as set forth in the remainder of the specification and terms and phrases in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description.

3. Claims 6, 17 and 41 are objected to because of the following informalities:

Amended Claim 6, Line 4, "among a plurality of threads on a result of the monitoring" appears to be incorrect and it appears that it should be "among a plurality of threads as a result of the monitoring".

Amended Claim 17, Lines 4-5, "among a plurality of threads on a result of the monitoring" appears to be incorrect and it appears that it should be "among a plurality of threads as a result of the monitoring".

Amended Claim 41, Line 5, "requesting a resource requesting in which a thread manager" appears to be incorrect and it appears that it should be "requesting a resource in which a thread manager".

Amended Claim 41, Lines 23-24, "outputting a result of the comparison in the comparison to an external unit" appears to be incorrect and it appears that it should be "outputting a result of the comparison to an external unit".

Appropriate correction is required.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

5. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. Claims 1, 3, 6, 13, 14, 17, 32 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent

6,345,242), and further in view of **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175).

6.1 **Chen** teaches multithreaded, mixed hardware description language logic simulation on engineering workstations. Specifically as per claim 1, **Chen** teaches a method of simulating an operation of a logical unit (CL1, L1-3; CL1, L12-15; CL3, L10-11; CL3, L18-20).

Chen teaches a thread manager, which controls threads each forming an execution unit of a program, for execution of each of threads representative of a series of functions required until the operation of the logical unit reaches completion according to a design specification of the logical unit (Fig 8 and Fig. 11). **Chen** does not expressly teach requesting a resource in which a thread manager makes a request for a hardware resource needed for execution of each of threads, to a resource manager which manages the hardware resource. **Dearth et al.** ('242) teaches requesting a resource in which a thread manager makes a request for a hardware resource needed for execution of each of threads, to a resource manager which manages the hardware resource (Fig. 2, Item 202 and Item 130; Abstract L17-20), because as per **Dearth et al.** ('824), when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('242) that included requesting a

resource in which a thread manager makes a request for a hardware resource needed for execution of each of threads, to a resource manager which manages the hardware resource. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach allocating a resource in which the resource manager allocates the hardware resource meeting the request to the thread in accordance with a rule prescribed in advance. **Dearth et al.** ('824) teaches allocating a resource in which the resource manager allocates the hardware resource meeting the request to the thread in accordance with a rule prescribed in advance (CL2, L46-48), because when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('824) that included allocating a resource in which the resource manager allocates the hardware resource meeting the request to the thread in accordance with a rule prescribed in advance. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be

necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen teaches controlling a thread in which the thread manager controls an execution state of the thread, the thread manager executing the requesting, allocating, and controlling repeatedly until the execution of the thread reaches completion, for simulating the operation of the logical unit to be conducted up to the completion (Fig 8 and Fig. 11). **Chen** does not expressly teach controlling a thread in which the thread manager controls an execution state of the thread in accordance with a result of the allocation made by the resource manager, the thread manager and the resource manager executing the requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of the thread reaches completion simulating the operation of the logical unit to be conducted up to the completion. **Dearth et al.** ('242) teaches controlling a thread in which the thread manager controls an execution state of the thread in accordance with a result of the allocation made by the resource manager, the thread manager and the resource manager executing the requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of the thread reaches completion simulating the operation of the logical unit to be conducted up to the completion (Fig. 2; Abstract L22-26), because as per **Dearth et al.** ('824), when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen**

with the method of **Dearth et al.** ('242) that included controlling a thread in which the thread manager controls an execution state of the thread in accordance with a result of the allocation made by the resource manager, the thread manager and the resource manager executing the requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of the thread reaches completion simulating the operation of the logical unit to be conducted up to the completion. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed. **Levy et al.** teaches dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed (CL3, L11-15; CL3, L22-30; CL3, L35-45), because that would support out-of-order execution of instructions for the threads (CL3, L14-15) with dynamic scheduling of instructions thus improving the performance of multithreaded processor (CL1, L38-41). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Levy et al.** that included dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed. The artisan would have been motivated because that would support out-of-order execution of instructions for the threads with dynamic scheduling of instructions thus improving the performance of multithreaded processor.

6.2 As per claim 3, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** teaches that the series of functions are represented in a plurality of sequential or concurrently executed threads (Fig. 8 and Fig 11).

6.3 As per claim 6, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the resource manager monitors resource requests in the requesting a resource to make a decision on a resource request deadlock state among a plurality of threads on a result of the monitoring. **Dearth et al.** ('824) teaches that the resource manager monitors resource requests in the requesting a resource to make a decision on a resource request deadlock state among a plurality of threads on a result of the monitoring (Abstract, L3-7; CL2, L30-33; CL3, L1-6), because when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid deadlocks in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('824) that included the resource manager monitoring resource requests in the requesting a resource to make a decision on a resource request deadlock state among a plurality of threads on a result of the monitoring. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the

simulated circuit or device; and it would be necessary to avoid deadlocks in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

6.4 As per claim 13, **Chen** teaches an apparatus for simulating an operation of a logical unit (CL1, L1-3; CL1, L12-15; CL3, L10-11; CL3, L18-20); comprising:

Chen teaches a thread manager for controlling a thread forming an execution unit of a program (Fig 8 and Fig. 11).

Chen does not expressly teach a resource manager for managing a hardware resource needed for execution of the thread; and resource allocating means for allocating a hardware resource meeting the request to the thread in accordance with a rule prescribed in advance. **Dearth et al.** ('824) teaches a resource manager for managing a hardware resource needed for execution of the thread; and resource allocating means for allocating a hardware resource meeting the request to the thread in accordance with a rule prescribed in advance (CL2, L46-48), because when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the apparatus of **Chen** with the apparatus of **Dearth et al.** ('824) that included a resource manager for managing a hardware resource needed for execution of the thread; and resource allocating means for allocating a hardware resource meeting the request to

the thread in accordance with a rule prescribed in advance. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen teaches a thread manager for execution of a thread representative of functions required until the operation of the logical unit reaches completion according to a design specification of the logical unit (Fig 8 and Fig. 11). **Chen** does not expressly teach resource requesting means for making a request for a hardware resource needed for execution of a thread to the resource manager. **Dearth et al.** ('242) teaches resource requesting means for making a request for a hardware resource needed for execution of a thread to the resource manager (Fig. 2, Item 202 and Item 130; Abstract L17-20), because as per **Dearth et al.** ('824), when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the apparatus of **Chen** with the apparatus of **Dearth et al.** ('242) that included resource requesting means for making a request for a hardware resource needed for execution of a thread to the resource manager. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be

necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen teaches thread control means for controlling an execution state of the thread; and the thread manager conducting the control of the thread execution state repeatedly in cooperation with each other until the execution of the thread reaches completion, for simulating the operation of the logical unit to be conducted up to the completion (Fig 8 and Fig. 11). **Chen** does not expressly teach thread control means for controlling an execution state of the thread in accordance with a result of a resource allocation made by the resource manager in response to the request from the resource requesting means; and the thread manager and the resource manager conducting the resource request and the control of the thread execution state repeatedly in cooperation with each other. **Dearth et al.** ('242) teaches thread control means for controlling an execution state of the thread in accordance with a result of a resource allocation made by the resource manager in response to the request from the resource requesting means; and the thread manager and the resource manager conducting the resource request and the control of the thread execution state repeatedly in cooperation with each other (Fig. 2; Abstract L22-26), because as per **Dearth et al.** ('824), when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the apparatus of **Chen** with the apparatus of **Dearth et al.** ('242) that included thread control means for controlling an execution state of the

thread in accordance with a result of a resource allocation made by the resource manager in response to the request from the resource requesting means; and the thread manager and the resource manager conducting the resource request and the control of the thread execution state repeatedly in cooperation with each other. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed. **Levy et al.** teaches dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed (CL3, L11-15; CL3, L22-30; CL3, L35-45), because that would support out-of-order execution of instructions for the threads (CL3, L14-15) with dynamic scheduling of instructions thus improving the performance of multithreaded processor (CL1, L38-41). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Levy et al.** that included dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed. The artisan would have been motivated because that would support out-of-order execution of instructions for the threads with dynamic scheduling of instructions thus improving the performance of multithreaded processor.

6.5 As per Claim 14, it is rejected based on the same reasoning as Claim 13, supra. Claim 14 is a computer readable recording medium claim reciting the same limitations as Claim 13, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.**

6.6 As per Claims 17 and 35, these are rejected based on the same reasoning as Claim 6, supra. Claims 17 and 35 are computer readable recording medium claims reciting the same limitations as Claim 6, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.**

6.7 As per Claim 32, it is rejected based on the same reasoning as Claim 3, supra. Claim 32 is a computer readable recording medium claim reciting the same limitations as Claim 3, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.**

7. Claims 2, 23 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Kinzelman et al.** (U.S. Patent 5,594,741).

7.1 As per claim 2, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the series of functions are represented in a plurality of sequential threads. **Kinzelman et al.** teaches that the series of functions are represented in a plurality of sequential threads (CL8, L49-52; CL11, L13-17), because that

allows instruction threads to be synchronized to align responder instructions from one transactor with the appropriate commander instructions (CL8, L46-49). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Kinzelman et al.** that included the series of functions being represented in a plurality of sequential threads. The artisan would have been motivated because that would allow instruction threads to be synchronized to align responder instructions from one transactor with the appropriate commander instructions.

7.2 As per claim 23, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the computer readable medium of claim 14. **Chen** does not expressly teach that the series of functions are represented in a plurality of sequential threads. **Kinzelman et al.** teaches that the series of functions are represented in a plurality of sequential threads (CL8, L49-52; CL11, L13-17). The motivation for combining **Chen** with **Kinzelman et al.** is presented in Paragraph 7.1 above.

7.3 As per claim 26, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. **Claim 26** has same limitations as Claim 17. The motivations for combining **Chen** with other references are presented in Paragraph 6.6 above.

8. Claims 4, 5, 15, 16, 33 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et**

al. (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **De Yong et al.** (U.S. Patent 5,355,435).

8.1 As per claim 4, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that a plurality of resource managers each corresponding to the resource manager are provided in conjunction with the types of the hardware resources, and in the allocating a resource, each of the resource managers allocates the hardware resource, the resource manager manages, to the thread in accordance with a local rule described in advance. **Dearth et al.** ('824) teaches that a resource manager is provided in conjunction with the types of the hardware resources, and in the allocating a resource, the resource manager allocates the hardware resource, the resource manager manages, to the thread in accordance with a local rule described in advance (Abstract, L1-9; CL1, L40-44; CL2, L46-48), because when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('824) that included a resource manager being provided in conjunction with the types of the hardware resources, and in the allocating a resource, the resource manager allocating the hardware resource, the resource manager managed, to the thread in accordance with a local rule described in advance. The artisan would have been motivated because when simulating a circuit using

multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach a plurality of resource managers each corresponding to the resource manager are provided in conjunction with the types of the hardware resources. **De Yong et al.** teaches a plurality of resource managers each corresponding to the resource manager are provided in conjunction with the types of the hardware resources (CL19, L35-36), because a plurality of hierarchical resource managers (arbitration systems) provide an ordered resolution of temporal contentions (CL19, L35-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **De Yong et al.** that included a plurality of resource managers each corresponding to the resource manager are provided in conjunction with the types of the hardware resources. The artisan would have been motivated because a plurality of hierarchical resource managers (arbitration systems) would provide an ordered resolution of temporal contentions.

8.2 As per claim 5, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach a plurality of resource managers each corresponding to the resource manager are provided in conjunction with the types of the hardware resources and are hierarchized according to the dependence among the hardware resources, and in the resource allocating, the hardware resource allocation is made in consideration of the dependence between the hardware resource managed by one of the resource

managers and the hardware resource managed by the other resource manager lower in hierarchy than the one of the resource managers. **De Yong et al.** teaches a plurality of resource managers each corresponding to the resource manager are provided in conjunction with the types of the hardware resources and are hierarchized according to the dependence among the hardware resources, and in the resource allocating, the hardware resource allocation is made in consideration of the dependence between the hardware resource managed by one of the resource managers and the hardware resource managed by the other resource manager lower in hierarchy than the one of the resource managers (CL19, L35-36), because a plurality of hierarchical resource managers (arbitration systems) provide an ordered resolution of temporal contentions (CL19, L35-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **De Yong et al.** that included a plurality of resource managers each corresponding to the resource manager were provided in conjunction with the types of the hardware resources and were hierarchized according to the dependence among the hardware resources, and in the resource allocating, the hardware resource allocation was made in consideration of the dependence between the hardware resource managed by one of the resource managers and the hardware resource managed by the other resource manager lower in hierarchy than the one of the resource managers. The artisan would have been motivated because a plurality of hierarchical resource managers (arbitration systems) would provide an ordered resolution of temporal contentions.

8.3 As per Claims 15, 16, 33 and 34, these are rejected based on the same reasoning as Claims 4 and 5, supra. Claims 15, 16, 33 and 34 are a computer readable recording medium

claim reciting the same limitations as Claims 4 and 5, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **De Yong et al.**

9. Claims 7, 18 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Thekkath et al.** (U.S. Patent 6,490,642).

9.1 As per claim 7, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the resource manager monitors read/write requests with respect to the hardware resource allocated by the resource request in the requesting a resource to make a decision on a competition state in read/write operation on the hardware resource among a plurality of threads on the basis of a result of the monitoring. **Dearth et al.** ('824) teaches that the resource manager monitors requests with respect to the hardware resource allocated by the resource request in the requesting a resource to make a decision on a competition state in operation on the hardware resource among a plurality of threads on the basis of a result of the monitoring (Abstract, L1-3; CL2, L30-33), because when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the

method of **Chen** with the method of **Dearth et al.** ('824) that included the resource manager monitoring requests with respect to the hardware resource allocated by the resource request in the requesting a resource to make a decision on a competition state in operation on the hardware resource among a plurality of threads on the basis of a result of the monitoring. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid deadlocks in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach that the resource manager monitors read/write requests with respect to the hardware resource allocated by the resource request in the resource requesting step to make a decision on a competition state in read/write operation on the hardware resource among a plurality of threads on the basis of a result of the monitoring. **Thekkath et al.** teaches that the resource manager monitors read/write requests with respect to the hardware resource allocated by the resource request in the resource requesting step to make a decision on a competition state in read/write operation on the hardware resource among a plurality of threads on the basis of a result of the monitoring (Abstract, L8-18; CL2, L23-34; CL2, L48-64; CL6, L65 to CL7, L5; CL9, L53-59), because that allows improving the efficiency of data transfers between devices interconnected over a system bus in a multi-master computer system configuration (Abstract, L1-4). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Thekkath et al.** that included the resource manager monitoring read/write requests with respect to the

hardware resource allocated by the resource request in the resource requesting step to make a decision on a competition state in read/write operation on the hardware resource among a plurality of threads on the basis of a result of the monitoring. The artisan would have been motivated because that would allow improving the efficiency of data transfers between devices interconnected over a system bus in a multi-master computer system configuration.

9.2 As per Claims 18 and 36, it is rejected based on the same reasoning as Claim 7, supra. Claims 18 and 36 are computer readable recording medium claims reciting the same limitations as Claim 7, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Thekkath et al.**

10. Claims 8, 10, 19, 21, 37 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Markov** (U.S. Patent 6,314,552).

10.1 As per claim 8, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the resource manager monitors the number of resource requests with respect to the hardware resource to detect a bottleneck on the thread on the basis of a result of the monitoring. **Markov** teaches that the resource manager monitors the number of resource requests with respect to the hardware resource to detect a bottleneck on the thread on the basis of a result of the monitoring (CL7, L27-34), because that

allows the resource manager to intervene and control the bottlenecks and allows evolutionary generation of candidate architectures (CL6, L7-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Markov** that included the resource manager monitoring the number of resource requests with respect to the hardware resource to detect a bottleneck on the thread on the basis of a result of the monitoring. The artisan would have been motivated because that would allow the resource manager to intervene and control the bottlenecks and allow evolutionary generation of candidate architectures.

10.2 As per claim 10, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the thread has a budget on a time of occupancy of a hardware resource allocated by the resource manager. **Markov** teaches that the thread has a budget on a time of occupancy of a hardware resource allocated by the resource manager (CL7, L27-34), because that allows the resource manager to intervene and control the time of occupancy and allows evolutionary generation of candidate architectures (CL6, L7-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Markov** that included the thread having a budget on a time of occupancy of a hardware resource allocated by the resource manager. The artisan would have been motivated because that would allow the resource manager to intervene and control the time of occupancy and allow evolutionary generation of candidate architectures.

10.3 As per Claims 19, 21, 37 and 39, these are rejected based on the same reasoning as Claims 8 and 10, supra. Claims 19, 21, 37 and 39 are computer readable recording medium claims reciting the same limitations as Claims 8 and 10, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Markov**.

11. Claims 9, 20 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Markov** (U.S. Patent 6,314,552) and **Kasuya** (U.S. Patent 6,077,304).

11.1 As per claim 9, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the resource manager monitors the number of resource requests with respect to the hardware resource to detect blocking of the resource requests on the basis of a result of the monitoring. **Kasuya** teaches that the resource manager monitors the number of resource requests with respect to the hardware resource to detect blocking of the resource requests on the basis of a result of the monitoring (Abstract, L12-16), because as per **Markov** that allows the resource manager to intervene and control the blocking and allows evolutionary generation of candidate architectures (CL6, L7-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Kasuya** that included the resource manager monitoring the number of resource requests with respect to the hardware resource to detect blocking of the resource requests on the basis of a result of the monitoring. The artisan would

have been motivated because that would allow the resource manager to intervene and control the blocking and allow evolutionary generation of candidate architectures.

11.2 As per Claims 20 and 38, these are rejected based on the same reasoning as Claim 9, supra. Claims 20 and 38 are computer readable recording medium claims reciting the same limitations as Claim 9, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al., Markov** and **Kasuya.**

12. Claims 11, 22 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Furuichi** (U.S. Patent 5,437,037).

12.1 As per claim 11, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824) and **Levy et al.** teach the method of claim 1. **Chen** does not expressly teach that the thread has an execution time-limit on the function. **Furuichi** teaches that the thread has an execution time-limit on the function (CL2, L34-36), because as per **Dearth et al.** ('824) collision in access to a simulated processor is avoided by reserving the simulated device to the concurrently executing threads for specific time period (Abstract, L1-3; CL2, L30-33). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Furuichi** that included the thread having an execution time-limit on the function. The artisan would have been motivated because that would allow collision in access to a simulated processor

to be avoided by reserving the simulated device to the concurrently executing threads for specific time period.

12.2 As per Claims 22 and 40, these are rejected based on the same reasoning as Claim 11, supra. Claims 22 and 40 are computer readable recording medium claims reciting the same limitations as Claim 11, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Furuichi.**

13. Claims 12 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Hollander** (U.S. Patent 6,347,388).

13.1 As per claim 12, **Chen** teaches a method of simulating an operation of a logical unit (CL1, L1-3; Cl1, L12-15; Cl3, L10-11; Cl3, L18-20).

Chen teaches a thread manager, which controls threads each forming an execution unit of a program, for execution of each of a series of threads representative of functions required until the operation of the logical unit reaches completion according to a design specification of the logical unit (Fig 8 and Fig. 11). **Chen** does not expressly teach requesting a resource in which a thread manager makes a request for a hardware resource needed for execution of each of a series of threads, to a resource manager which manages the hardware resource. **Dearth et al.** ('242)

teaches requesting a resource in which a thread manager makes a request for a hardware resource needed for execution of each of a series of threads, to a resource manager which manages the hardware resource (Fig. 2, Item 202 and Item 130; Abstract L17-20), because as per **Dearth et al.** ('824), when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('242) that included requesting a resource in which a thread manager makes a request for a hardware resource needed for execution of each of a series of threads, to a resource manager which manages the hardware resource. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach allocating a resource in which the resource manager allocates the hardware resource meeting the request to the thread in accordance with a rule prescribed in advance. **Dearth et al.** ('824) teaches allocating a resource in which the resource manager allocates the hardware resource meeting the request to the thread in accordance with a rule prescribed in advance (CL2, L46-48), because when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same

simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('824) that included allocating a resource in which the resource manager allocates the hardware resource meeting the request to the thread in accordance with a rule prescribed in advance. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen teaches controlling a thread in which the thread manager controls an execution state of the thread, the thread manager executing the requesting, allocating, and controlling repeatedly until the execution of the thread reaches completion, for simulating the operation of the logical unit to be conducted up to the completion (Fig 8 and Fig. 11). **Chen** does not expressly teach controlling a thread in which the thread manager controls an execution state of the thread in accordance with a result of the allocation made by the resource manager, the thread manager and the resource manager executing the requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of the thread reaches completion simulating the operation of the logical unit to be conducted up to the completion. **Dearth et al.** ('242) teaches controlling a thread in which the thread manager controls an execution state of the thread in accordance with a result of the allocation made by the resource manager, the thread

manager and the resource manager executing the requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of the thread reaches completion simulating the operation of the logical unit to be conducted up to the completion (Fig. 2; Abstract L22-26), because as per **Dearth et al.** ('824), when simulating a circuit using multiple concurrently executing threads, often more than one thread attempts to interact with the same simulated component of the simulated circuit or device; and it is necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads (CL2, L30-33; Abstract, L1-3; CL4, L32-36). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Dearth et al.** ('242) that included controlling a thread in which the thread manager controls an execution state of the thread in accordance with a result of the allocation made by the resource manager, the thread manager and the resource manager executing the requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of the thread reaches completion simulating the operation of the logical unit to be conducted up to the completion. The artisan would have been motivated because when simulating a circuit using multiple concurrently executing threads, often more than one thread would attempt to interact with the same simulated component of the simulated circuit or device; and it would be necessary to avoid collisions in multiple concurrently executing threads by reserving simulated devices to one or more simulation threads.

Chen does not expressly teach dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed. **Levy et al.** teaches dynamically allocating necessary hardware resources to the thread by the resource

manager every time the generated thread is executed (CL3, L11-15; CL3, L22-30; CL3, L35-45), because that would support out-of-order execution of instructions for the threads (CL3, L14-15) with dynamic scheduling of instructions thus improving the performance of multithreaded processor (CL1, L38-41). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Levy et al.** that included dynamically allocating necessary hardware resources to the thread by the resource manager every time the generated thread is executed. The artisan would have been motivated because that would support out-of-order execution of instructions for the threads with dynamic scheduling of instructions thus improving the performance of multithreaded processor.

Chen does not expressly teach comparing a result of the simulation with an estimated value on the operation of the logical unit. **Hollander** teaches comparing a result of the simulation with an estimated value on the operation of the logical unit (CL1, L53-55; CL1, L66-67), because that allows the designer o determine whether a particular hardware and software combination exactly implements the requirements defined by the IC's specification (CL1, L15-18). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Hollander** that included comparing a result of the simulation with an estimated value on the operation of the logical unit. The artisan would have been motivated because that would allow the designer to determine whether a particular hardware and software combination exactly implements the requirements defined by the IC's specification.

Chen does not expressly teach outputting a result of the comparison to an external unit. **Hollander** teaches outputting a result of the comparison to an external unit (CL2, L25-27; CL2,

L33-34), because that allows the designer to determine whether a particular hardware and software combination exactly implements the requirements defined by the IC's specification (CL1, L15-18). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Chen** with the method of **Hollander** that included outputting a result of the comparison to an external unit. The artisan would have been motivated because that would allow the designer to determine whether a particular hardware and software combination exactly implements the requirements defined by the IC's specification.

13.2 As per Claim 41, it is rejected based on the same reasoning as Claim 12, supra. Claim 41 is a computer readable recording medium claim reciting the same limitations as Claim 11, as taught throughout by **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Hollander.**

14. Claims 24 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Kinzelman et al.** (U.S. Patent 5,594,741) and **De Yong et al.** (U.S. Patent 5,355,435)

14.1 As per claim 24, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. Claim 24 has same limitation as claim 15, which is taught by **De Yong et al.** The motivation for combining **Chen** with **De Yong et al.** is presented in Paragraph 8.1 above.

14.2 As per claim 25, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. Claim 25 has same limitation as claim 16, which is taught by **De Yong et al.** The motivation for combining **Chen** with **De Yong et al.** is presented in Paragraph 8.2 above.

15. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Kinzelman et al.** (U.S. Patent 5,594,741) and **Thekkath et al.** (U.S. Patent 6,490,642).

15.1 As per claim 27, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. Claim 27 has same limitation as claim 18, which is taught by **Thekkath et al.** The motivation for combining **Chen** with **Thekkath et al.** is presented in Paragraph 16.1 above.

16. Claims 28 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), and **Dearth et al.** (U.S. Patent 5,812,824), and further in view of **Kinzelman et al.** (U.S. Patent 5,594,741) and **Markov** (U.S. Patent 6,314,552).

16.1 As per claims 28 and 30, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. claims 28 and 30 have same limitations as claim 8 and 10, which are taught by **Markov.** The motivations for combining **Chen** with **Markov** are presented in Paragraphs 10.1 and 10.2 above.

17. Claim 29 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Kinzelman et al.** (U.S. Patent 5,594,741), **Markov** (U.S. Patent 6,314,552) and **Kasuya** (U.S. Patent 6,077,304).

17.1 As per claim 29, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. Claim 29 has same limitation as claim 20, which is taught by **Markov** and **Kasuya.** The motivation for combining **Chen** with **Markov** and **Kasuya** is presented in Paragraph 11.1 above.

18. Claim 31 is are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chen** (U.S. Patent 6,466,898) in view of **Dearth et al.** (U.S. Patent 6,345,242), **Dearth et al.** (U.S. Patent 5,812,824) and **Levy et al.** (U.S. Patent 6,092,175), and further in view of **Kinzelman et al.** (U.S. Patent 5,594,741) and **Furuichi** (U.S. Patent 5,437,037).

18.1 As per claim 31, **Chen, Dearth et al.** ('242), **Dearth et al.** ('824), **Levy et al.** and **Kinzelman et al.** teach the computer readable medium of claim 23. Claim 31 has same

limitation as claim 22, which is taught by **Furuichi.** The motivation for combining **Chen** with **Furuichi** is presented in Paragraph 19.1 above.

Response to Arguments

19. Applicants' arguments filed on May 10, 2005 and July 29, 2005 have been fully considered. The arguments with respect to 103 (a) rejections are moot, in view of new rejections made against the amended claims.

19.1 As per the applicants' argument that "the combination of the references merely teaches avoiding a collision of reservation requests for hardware resource tests only at a time of parallel execution of threads; none of the art, alone or in combination teaches, for example, a method, including "requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of said thread reaches completion while dynamically allocating necessary hardware resources to the thread by said resource manager every time the generated thread is executed, for simulating the operation of said logical unit to be conducted up to the completion"; the cited art, alone or in combination, teach away from a dynamic allocation by a resource manager in allocating necessary hardware resources every time a generated thread is executed; Dearth '242 merely teaches that execution of a user thread is suspended substantially immediately after registration of a test Dearth '824 merely teaches that when the subject test is granted reservation of all reserved devices, hub signals the subject test to awaken, i.e., to resume execution; ", the examiner has used a new reference **Levy et al.**

Levy et al. teaches requesting, allocating, and controlling repeatedly in cooperation with each other until the execution of said thread reaches completion while dynamically allocating necessary hardware resources to the thread by said resource manager every time the generated thread is executed, for simulating the operation of said logical unit to be conducted up to the completion (CL3, L11-15; CL3, L22-30; CL3, L35-45), because that would support out-of-order execution of instructions for the threads (CL3, L14-15) with dynamic scheduling of instructions thus improving the performance of multithreaded processor (CL1, L38-41).

Conclusion

ACTION IS FINAL

20. Applicant's amendments necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

21. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dr. Kandasamy Thangavelu whose telephone number is 571-272-3717. The examiner can normally be reached on Monday through Friday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard, can be reached on 571-272-3749. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Paul L. Rodriguez 8/25/05

Primary Examiner
Art Unit 2125

K. Thangavelu
Art Unit 2123
August 24, 2005